

PopED lite: an optimal design software for preclinical pharmacokinetic and pharmacodynamic studies

Yasunori Aoki ^{*1}, Monika Sundqvist², Andrew C. Hooker¹, and Peter Gennemark²

¹Pharmacometrics Research Group, Dept. Pharmaceutical Biosciences, Uppsala University,
Box 591, SE-751 24, Uppsala, Sweden

²CVMD iMed DMPK AstraZeneca R&D, SE-431 83 Mölndal, Sweden

May 26, 2015

Abstract

Optimal experimental design approaches are seldom used in pre-clinical drug discovery. Main reasons for this lack of use are that available software tools require relatively high insight in optimal design theory, and that the design-execution cycle of in vivo experiments is short, making time-consuming optimizations infeasible. We present the publicly available software PopED lite in order to increase the use of optimal design in pre-clinical drug discovery. PopED lite is designed to be simple, fast and intuitive. Simple, to give many users access to basic optimal design calculations. Fast, to fit the short design-execution cycle and allow interactive experimental design (test one design, discuss proposed design, test another design, etc). Intuitive, so that the input to and output from the software can easily be understood by users without knowledge of the theory of optimal design. In this way, PopED lite is highly useful in practice and complements existing tools. Key functionality of PopED lite is demonstrated by three case studies from real drug discovery projects.

Keywords: optimal experimental design, pre-clinical drug discovery, model-based drug discovery.

1 Introduction

Poorly designed experiments often lead to uninformative experimental results. For example, poorly chosen dosage and observation/sampling times in an animal experiment can lead to inaccurate characterisation of drug candidates. In order to minimize the risk of running uninformative experiments in drug discovery, we introduce the

^{*}Corresponding author: yaoki@uwaterloo.ca

software PopED lite that systematically incorporates a priori knowledge about the compound and algorithmically optimizes the experimental design.

Decisions in preclinical drug discovery are largely based on inference of data from cell-based assays and animals exposed to drug candidates. Accurate efficacy and safety estimates of each test compound are pivotal for proper compound ranking. Mathematical models are commonly used to formally integrate available information in order to gain system-level understanding of pharmacological effects and to make informed decisions [10, 20]. These models are usually composed of two parts; the pharmacokinetics (PK) part representing what the body does to the drug, and the pharmacodynamics (PD) part representing what the drug does to the body [9].

During the compound selection phase of a drug discovery project, several compounds are tested. Estimated model parameters (e.g., clearance and potency) are used to quantitatively characterise the compound. Naturally, the uncertainty of parameter estimates must be taken into account. The design of the experiment (e.g., when to take the blood samples, how much compound to administer) can have large influence on the parameter estimation uncertainty. Therefore, experimental design of preclinical studies is a crucial component of overall success in drug discovery.

Currently, in a typical preclinical study, experiments are manually designed and sampling times are often fixed for all test compounds. Although there have been significant developments in optimal design for clinical studies [8, 11, 12, 13, 14], the use of optimal design in preclinical studies is still limited [17, 18]. Thus, we hypothesise that there is significant room for improvement in the experimental design of preclinical studies. For example, the power of prior information from similar compounds is not fully taken into account.

Initially, we attempted to introduce already existing optimal design software [4, 14] into a realistic experimental design workflow of pre-clinical PK/PD studies. During this attempt, we learned how preclinical experiments were designed in drug discovery and identified several causes for optimal design not being used in pre-clinical studies. Specifically, we identified three key challenges in utilizing the idea of optimal design in preclinical experiments.

Firstly, the currently available experimental design tools are developed by academic groups with the aim of delivering state of the art optimal design theories to a wide variety of applications. Therefore these software tools are developed for high functionality and high flexibility. However, flexibility incurs more user inputs which can demotivate unexperienced users and increase the risk of user errors. As the development of experimental designs for similar preclinical studies is often done routinely, the high flexibility is not necessary and hinders users who are not familiar with optimal design software tools.

Secondly, most of the currently available software tools are designed to be used for clinical studies. For a clinical study the true optimality of the design is desirable, since study cost is high and since the design-experiment

cycles are in the order of months. As a consequence, currently available software tools sacrifice speed of computation for optimality of the design. However, for preclinical studies, the design-experiment cycles are in the order of weeks, and often the experimental designs are adjusted during discussions with experimentalists. Thus, an optimal design software that requires considerable computational time is not suitable for preclinical studies.

Thirdly, most of the optimal design software tools assume basic user knowledge in optimal design, and use optimal design terminology (e.g., D-optimal criterion, determinant of FIM,...) in the user interface to concisely communicate the idea behind the software. However, the majority of preclinical experimentalists are not used to optimal design terminology and interpretation of the required user-input and software-output of these programs becomes an issue.

To address these three challenges we have implemented the software PopED lite that is designed to be a simple, fast and intuitive playground for experimental design in the preclinical area of drug development. PopED lite is available on the Mac App Store <https://itunes.apple.com/se/app/poped-lite/id836277613?l=en&mt=12> for Mac and at http://www.bluetree.me/PopED_lite for Windows.

2 Design Considerations

Through our initial attempt of introducing pre-existing optimal design software tools to industrial drug-discovery teams, we observed that these were too complex for the relatively simple experimental design computations needed for preclinical PK/PD studies, too slow to fit into the compound-selection workflow, and too difficult to understand for users without prior knowledge in optimal design. PopED lite was designed to be simpler, faster and more intuitive compared to existing programs in order to bring optimal design into the regular workflow of preclinical animal experiments in the pharmaceutical industry. Together with an industrial drug-discovery team we conducted a retrospective study (Case Study 1) and with another team we designed experiments using a preliminary version of PopED lite (Case study 2) as well as with the current version (Case Study 3). Through these case studies in addition to the discussions with many other preclinical experimental teams, we have attempted to balance the appropriate flexibility, accuracy and level of technicality.

Flexibility. PopED lite focuses on optimization of the dosage and PK/PD sampling (observation) times to improve the accuracy of the parameter estimates of fixed effect PK/PD models. Pre-clinical experiments are characterised by relatively small group sizes, and animals with small genetic differences. The main interest is to predict the human situation, and in this translation focus is on average levels and not on variability, as translation of the latter is associated with very high uncertainty. Therefore, the use of nonlinear mixed effect modelling is not

as central as in the clinical setting, and key questions can often be addressed by standard deterministic (fixed effect) PK/PD models.

PopED lite implements standard compartmental PK models (1-, 2-, and 3-compartment models with linear and nonlinear elimination, and linear absorption), as well as commonly used PD models. For PD models, the user can alternatively input a regular mathematical expression in form of a function or an ordinary differential equation. This choice of model flexibility is chosen to cover a wide range of possible PK/PD models and at the same time ensure software usability.

Accuracy. We have decided to discretize both sampling time and dosage search space, so that the optimal design can be chosen from a combination of practical sampling schedules and allowed possible dosages.

We have observed that a precisely optimized experimental design is seldom practically useful. For example, in practice one can only sample every 5th minute and a discrete search space for sampling or dose times is sufficient. By discretizing the sampling time and dosage the design space will shrink to finite, and we can reduce the computational cost for the design optimization while avoiding unrealistically precise experimental designs.

We have also observed that a quickly obtained good-enough design, and not necessarily the optimal, is often more valuable in preclinical studies due to their time constraints.

Level of Technicality. Preclinical studies are designed by cross-functional teams (e.g., biologists, chemists, and PK/PD-modellers), that generally lack training in numerical computation and information theory. Therefore, optimal-design specific jargon should be avoided in the software interface, and PopED is designed to only display terminology that is familiar to these teams. Then based on the information specified by the user, PopED lite chooses an appropriate design optimization strategy. In this way, PopED lite introduces the idea of optimal design to preclinical experimentalists.

In summary, the design philosophy of PopED lite is to create the simplest, fastest, most intuitive optimal design software that maintains the appropriate flexibility, accuracy and level of technicality for use in preclinical drug discovery.

3 Material and Methods

In PopED lite, we obtain an optimized experimental design by solving a constrained optimization problem, where each experimental design is evaluated using some function of the Fisher Information Matrix (FIM). The FIM is calculated based on the PK/PD model structure, possible sets of model parameters, and the experimental design (see [3] for an introduction to the field).

3.1 Overall Workflow and User Interface

User input to PopED lite is entered step by step; completion of one step is required before the next step appears. This increases usability and reduces user errors. The optimal design is calculated in five steps (Figure 1). Users are only required to input numbers, and a single mathematical expression in case a user-specified PD model is desired.

3.2 Computation related to the Fisher Information Matrix

Accurate construction and computation of the FIM is vital to optimal design software, as the FIM is used to quantify the quality of the experimental design. In order to assure appropriate FIM construction, the residual error model is automatically built using the estimated limit of quantifications and measurement errors provided by the user in Step 2 (Figure 1) of PopED lite. In addition, since the FIM is a symmetric matrix PopED lite utilizes Cholesky decomposition to calculate the determinant and inverse of the FIM. At each Cholesky decomposition, PopED lite monitors the rounding error and also checks the invertibility of the FIM. If the FIM of the optimized design is singular, the program outputs a warning message saying that the parameters cannot be uniquely identified from the experiment, instead of stating that the FIM is singular. Naturally, this facilitates user interpretation. In addition, if the computational error is detected when inverting the FIM, PopED lite outputs a warning as well as reports the corresponding Coefficients of Variation (CVs) with a question mark (?), allowing the user to interpret these CVs with caution. Appendix A describes in detail how the FIM is constructed in PopED lite.

3.3 PK/PD model simulation

The majority of the computational cost for optimal design software originates from the computation of the FIM. This computational cost in calculating the FIM is reduced by hard-coding several PK and PD models in PopED lite. For user defined PD functions, PopED lite implements an expression parser based on the shunting-yard algorithm [6]. PopED lite parses the expression only once to save computation time. All subsequent evaluations of the expression in the design optimization procedure reuse the parsed expression. For PK/PD models that require solving a system of ordinary differential equations, we use the Rosenbrock method [16] implemented in the boost package in the Odint toolbox [1]. The Jacobian of the right hand side of a user defined PD function is approximated using a forward difference scheme.

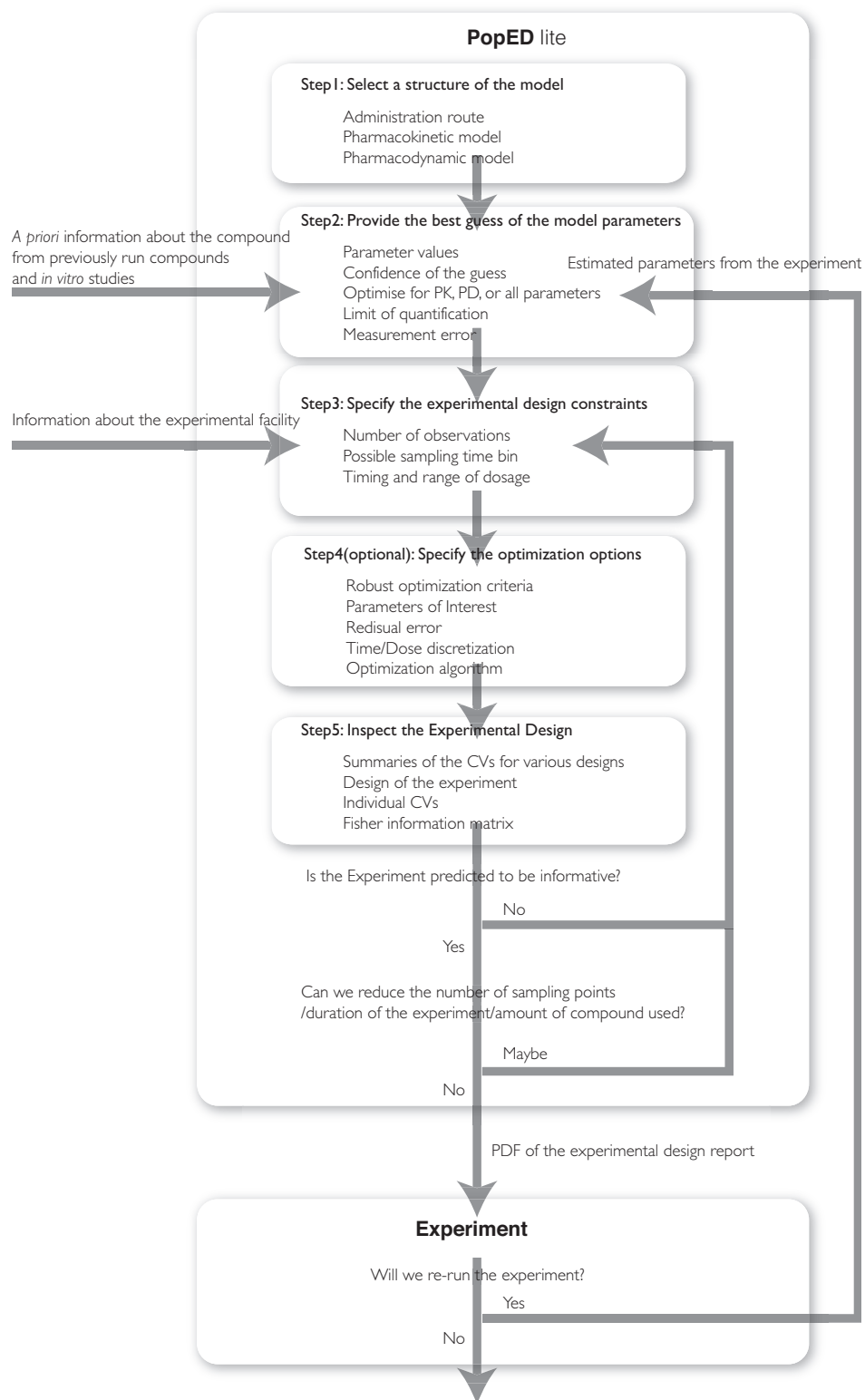


Figure 1: Overall workflow of PopED lite.

3.4 Optimization Criteria

PopED lite chooses the optimization criteria from D, Ds, ED, and EDs optimal designs [3, 7] to fit the optimization need of the user. For a chosen PK/PD model structure, the user can specify the confidence level of the best guess of the PK/PD parameters (Step 2 of PopED lite). If there is no uncertainty, PopED lite optimizes the experimental design using the D optimization criterion. Otherwise, it randomly generates sets of parameters following the uniform distribution in the range specified by the user and conducts ED optimal design (by taking the expectation of the natural logarithm of the determinant of the FIM for randomly generated sets of parameters). Furthermore, the user can specify in Step 2 that PopED lite should optimize the design for parameter estimation accuracy of a subset of parameters; Ds or EDs optimization criteria will then be used. Note that these optimization criteria do not appear in the standard user interface and PopED lite interprets the optimization needs of the user and chooses a criterion automatically.

3.5 Optimization Algorithms

We have implemented discrete optimization algorithms, in order to achieve the desired speed and also to obtain an experimental design that makes sense in practice.

By discretizing the sampling time, it is possible to reduce the search space (the design space) significantly. A key advantage of discretizing the sampling time is that model simulations are avoided during sampling time optimization since all the elements that can appear in the Jacobian matrices can be pre-calculated. This dramatically reduces the cost of computation especially when the model is defined by a system of ordinary differential equations (ODEs).

By discretizing the dosage, the user can specify the possible doses considering the experimental constraints and compound formulation constraints. As a result, in particular for single dose experiments, PopED lite can search exhaustively for the dosage and hence avoid the risk of finding a locally optimal design.

The sampling time optimization is a combinatorial optimization problem (e.g. choose 6 sampling points from 420 possible sampling points), while the dose optimization is a simple discrete optimization problem. PopED lite implements stochastic optimization algorithms for these two problems to reduce the risk of finding a local optimal design. The two algorithms are detailed in Appendix C.

3.6 Implementation Framework

PopED lite was implemented using the Qt framework [5] and overall implementation was optimized using Instruments [2]. The Qt framework was chosen, for its speed of computation, ease of deployment, and flexibility of GUI design. The Qt framework is based on C++, hence PopED lite takes advantage of the speed of compiled software. Most of the optimal design programs targeted for pharmaceutical drug development are written in interpreter based languages such as R and Matlab which is a major cause of slow computation. Typically one order of magnitude faster computation can be expected for compiler based language such as C++, C, or FORTRAN compared to interpreter based languages. The Qt is also a cross-platform framework so that PopED lite can be deployed to both Unix-based platforms and Windows-based platforms. We have also made PopED lite a standalone software to facilitate installation. As a result, PopED lite does not require any extra installation step and can be started by simply clicking on an icon just like any other Apps available on a computer. Lastly, the Qt framework supports flexible GUI design with a unified look and feel.

4 Case Studies

We now present the case studies that PopED lite was motivated by, developed for, and used in. All case studies are based on problems encountered in authentic drug discovery projects.

4.1 Case study 1: Designing a mouse receptor-occupancy study to estimate test-compound potency

This case study was run before the start of the development of PopED lite, and motivated the development of optimal design software specifically designed for preclinical studies. In a drug candidate optimization program, several chemically similar compounds were evaluated in the mouse. Specifically, time series data for drug exposure and drug targeted receptor occupancy were generated from orally dosed mice. For the latter time-series, one mouse was sacrificed for each sampling point making experimental design highly important both from ethical and cost perspectives. For the compound series, PK could be reliably modeled by a linear two-compartment model (1st order absorption, linear elimination), and the distribution between drug-receptor complex (RC) and free receptor (R) was modeled by a receptor kinetic model with elementary reactions as

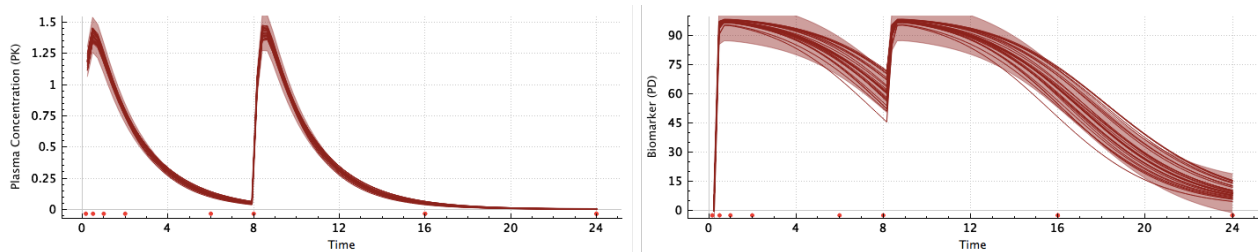
$$\frac{d}{dt}RC(t) = k_{\text{on}}C(t)(R_{\text{tot}} - RC(t)) - k_{\text{off}}RC(t)$$

where C denotes drug plasma concentration, R_{tot} denotes the total receptor concentration, RC denotes the drug-receptor complex and k_{on} and k_{off} are kinetic parameters. The dissociation constant is derived as $K_M = k_{\text{off}}/k_{\text{on}}$, and reflects the potency of the test compound. The in vivo experiment was routinely conducted using a manually defined standard design. For one test compound, the CV of k_{off} was as high as 80% resulting in high uncertainty in the potency estimate. The reason for this high CV was that all observations after 16 hr were below the limit of quantification and hence not informative. To improve the estimate, a second experiment was planned. During the experimental planning, various simulations were run with manually designed experiments and the experimental design with two doses of 40 $\mu\text{mol/kg}$ (at time 0 h and 8 h) and sampling at 8, 16, 16, 24, and 24 hrs was chosen for the second experiment (Figure 2(a)). By combining the first and the second experiments, the predicted CV for k_{off} was 52%. Actual data generated using this design reduced the CV of k_{off} even more, down to 22%, and the potency could be reliably derived.

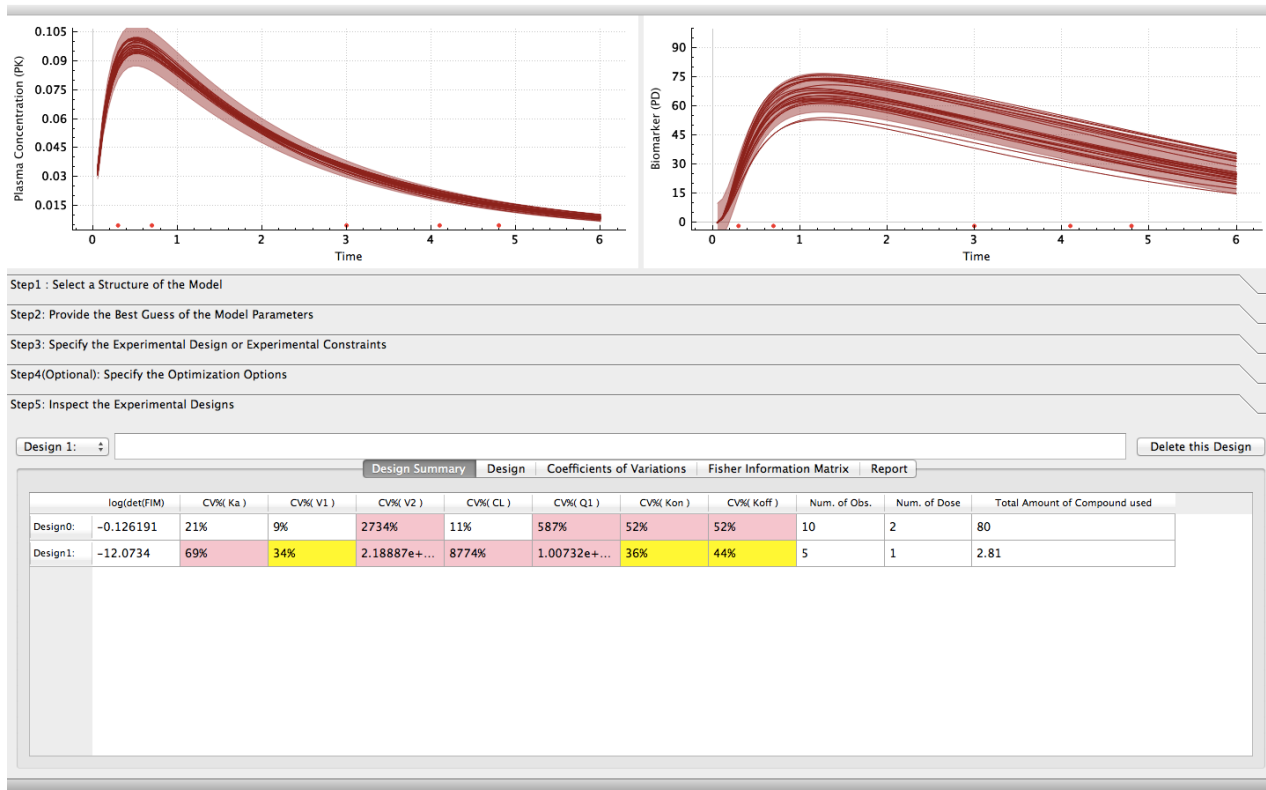
In a retrospective analysis we explain how this experiment could have been optimized even further using PopED lite. We first tried to optimize for the sampling schedule only, but did not find significant improvement on the predicted CV for k_{on} and k_{off} . In a subsequent step we optimized for the dosing amount using an optimization criterion taking the accuracy of all the model parameters into account (ED optimal design). The proposed design also failed to improve the CV for k_{off} and k_{on} . However, by optimizing only for the PD parameters (EDs optimal design), the predicted CVs for k_{off} and k_{on} were reduced to 36 and 44%. As the predicted CVs of k_{off} and k_{on} for the actual experiment ran were 52% and 52% (see Figure 2(b) design 0), we can expect further reduction in the parameter estimation uncertainty with this design. In addition, this optimal design requires a short sampling period (0.3, 0.7, 3, 4.1, and 4.8 hours), and only one dosing of 2.8 $\mu\text{mol/kg}$ at time 0.

By using EDs optimal design with PD parameters as the parameters of interest, it is not unexpected that the CVs of the PK parameters are relatively high in the proposed design. However, even if individual PK parameters cannot be identified accurately, the shape of the PK profile can be roughly estimated from the observation, and we can estimate PD parameters accurately using a relatively sparse sampling schedule.

In summary, the retrospective analysis indicates that by using an optimal design software we could have reduced the required amount of compound, which is often limited in preclinical studies, and study duration while obtaining similarly accurate parameter estimates. This example also illustrates the importance of both dose and sampling time optimization as well as optimization focused on the parameters of interest (i.e. EDs optimal design).



(a) PK (left) and PD (right) simulation of the manually designed experiment for Case study 1. Red dots represent the sampling time. Red lines are the simulations with random parameters generated from the parameter guesstimates and their uncertainties. The shaded area is the 68% prediction interval of the observation based on the measurement error, LOQ, and parameter guesstimate.



(b) Graphical user interface of PopED lite. Design0 in the table (the first line) is the manual design and Design1 is the design optimized by PopED lite.

Figure 2: Case study 1. PopED lite screenshot of the summary of the experimental designs. Design1 gives lower CV% for k_{off} and k_{on} and requires less number of observations and amount of compound compared to Design0.

4.2 Case study 2: Designing a dog study to estimate test compound potency

PopED lite was developed along with a drug discovery project and a preliminary version of PopED lite was used to design several dog experiments. The drug discovery project aimed at decreasing the level of a biomarker by more than 80% over 24 h. The compound concentration and biomarker response could be measured both in vitro and in vivo. However, several compounds displayed a time delay between concentration and effect in vivo that was not fully captured in the in vitro assay. To better understand the exposure-response relationship, the compound was administered to dogs in vivo, and exposure and biomarker was followed over time. The PK was then modeled using a 2-compartment model. The biomarker was modeled using a link model (1) coupled to an Emax model (2)

$$C_e = k_e(C_p - C_e), \quad (1)$$

$$E = 1 - \frac{C_e}{IC_{50} - C_e} \quad (2)$$

where C_e denotes the effective concentration, k_e denotes the equilibrium constant, C_p denotes the plasma concentration, E denotes the biomarker concentration and IC_{50} signifies the concentration required to reach 50% inhibition (Emax was assumed to be one). Because of the time delay between concentration and response, it is not intuitively easy to propose sampling time-points that result in informative data. To increase the accuracy of the parameter estimates a preliminary version of PopED lite was used.

The sampling times of the first experiment was optimized based on the model parameters from in vitro data and previously run compounds in the chemical series. The predicted CV for IC_{50} for the optimized design was 25%. With the preliminary version of PopED lite, we were only able to conduct Ds optimal design; uncertainty of the initial guess of the parameters was not considered in the design. The experiment was run with the optimized sampling time and fixed dosage of 0.44 $\mu\text{mol/kg}$; however, most of the PD measurements were below the limit of quantification and IC_{50} could therefore not be reliably estimated (CV of IC_{50} was over 400%). The imperfect experimental design was due to a poor estimate of IC_{50} based on in vitro data; about one order of magnitude different from that estimated from in vivo data. It was clear that the dosage should be reduced in the next experiment in order to avoid PD measurements below the limit of quantification; however, it was not clear by how much.

In the next experiment, we used a preliminary version of PopED lite to optimize the dose based on the estimated parameters from the first experiment. The preliminary version of PopED lite had a functionality to generate feasible sets of model parameters (data import followed by bootstrapping) and then run ED(s) optimal

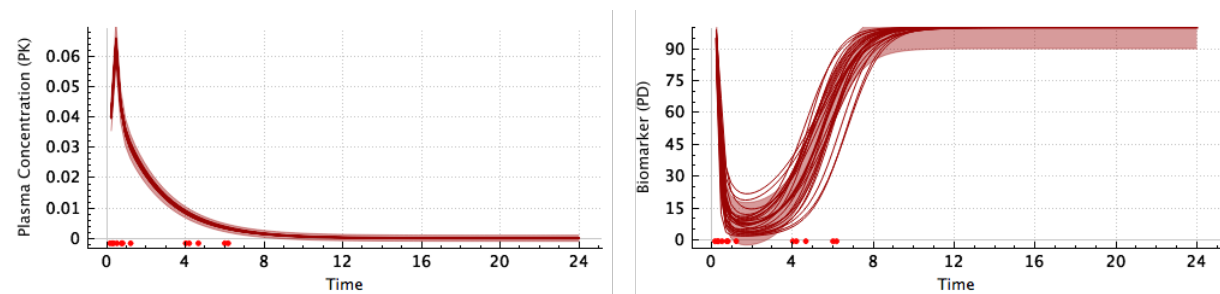


Figure 3: Case Study2: Optimized experimental design using the estimated parameters from the first experiment.

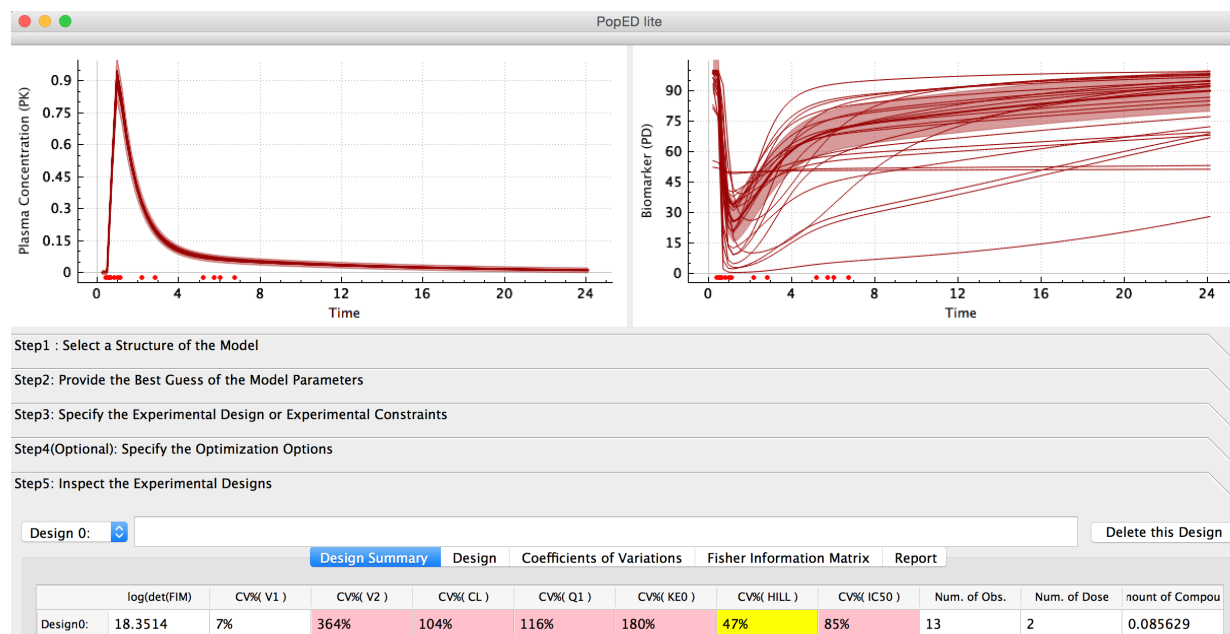


Figure 4: Retrospective study of Case study 2

design based on these sets. The preliminary version of PopED lite suggested the new dose to be $0.04 \mu\text{mol/kg}$, and predicted the CV of the IC_{50} to be 10%. The experiment was run with the design suggested by PopED lite and the resulting IC_{50} estimate had a CV of 7%.

Similar analysis can be done with the current version of PopED lite, i.e., with 5% uncertainties in PK parameters and 30% uncertainties in PD parameters (Figure 3).

As a retrospective study we included 100% uncertainty to the prior guess of the PD parameters, based on the in vitro data. As can be seen in Figure 4, using the EDs optimal criterion PopED lite suggested a dose of $0.08 \mu\text{mol/kg}$. Hence, we could have avoided the uninformative first experiment if the uncertainty of the initial guess of the parameters had been incorporated appropriately in the optimization of the experimental design. From this case study, we see that a poor initial guess of the parameter values leads to a poor experimental design.

To avoid this problem, uncertainty in the prior guess of the parameters should be included in the experimental design calculation. Based on this case study, we have made ED/EDs optimal design to be the default setting for PopED lite requiring the user to specify the confidence level of the initial guess of the parameters.

4.3 Case study 3: Designing a mouse study to investigate renal function

The last case study describes how the deployed version of PopED lite was proven to be a useful tool for motivating an experimental design that was different from the standard design of a preclinical study. The purpose of the study was to investigate renal function in an in vivo mouse model by measuring the glomerular filtration rate (GFR). The standard protocol for GFR measurement is to study clearance of a metabolically inert compound (inulin or sinestrin) that is freely filtered, not bound to plasma proteins and not subject to reabsorption [19]. The compound is administered intravenously and the plasma concentration is followed over time. The clearance estimate is subsequently obtained by fitting a standard two compartment PK model to the concentration-curve.

In an initial study, a bolus intravenous injection of sinestrin, and standard sampling-points (3, 7, 10, 15, 35, 55, 75 min) found in literature were used [15]. However, the compound had an extremely rapid distribution phase in plasma and only approximately 40% of the mice showed plasma curves that were possible to fit to a two-compartment model.

Using PopED lite, the impact of changing the sampling time on the certainty of the estimates could easily be demonstrated (Figure 5). If sampling was restrained to later than 3 minutes the uncertainty in the parameters was much larger than if a sample could be taken at less than one minute. Specifically, we first obtained rough estimates of the PK parameters from previous ran studies and literature values ($V_1=3000 \mu\text{l}$, $V_2=5000 \mu\text{l}$, $CL=300 \mu\text{l}/\text{min}$, and $Q_1=1000 \mu\text{l}/\text{min}$), and estimated the uncertainty of these rough estimates to be plus or minus 30%. Then, we evaluated the standard design by calculating the predicted CVs. The CV of CL was predicted to be approximately 38% for this study design. We then calculated the optimal experimental design under the design constraint that seven samples could be taken any minute between 0 min and 75 min. For the new design (1, 2, 5, 10, 35, 75 min), the CV of CL was predicted to reduce to 4%. Although the proposed design deviated from the standard protocol, the PopED lite calculations convinced the drug discovery team to run the new design. The resulting measurements from three mice were then modeled with a two-compartment model and the estimated CL had a CV of only 2%. Through this successful integration of PopED lite to a preclinical experimental design workflow we showed that even for a very rough initial estimate of the parameters, the experimental design can be improved significantly.



Figure 5: Case study 3. PopED lite screenshot of the summary of the experimental designs. Design0 is the standard design and Design1 is the design optimized by PopED lite. All CVs are predicted to decrease significantly in the optimized design.

5 Discussion and Concluding Remarks

In this paper we have demonstrated how the optimal design software PopED lite was designed collaboratively by academic researchers and preclinical experimental teams in the pharmaceutical industry. It is designed to fit into a practical workflow, so that the use of an optimal design software can be the standard procedure in preclinical experiment designing and thus help to increase the efficiency of drug discovery in vivo studies.

Our initial attempt was to introduce the already existing software PopED and create a simplified GUI for pre-clinical experiments. However, as we investigated the preclinical experimental design workflow in more detail, we realised that a significant simplification to the user-software interaction, a thorough optimization of the computational speed, and a dramatic reduction of the level of technicality of the software were required. To address these issues we decided to reconstruct the software from scratch. This has allowed us to design the software-user interaction first and then implement the numerical computation engines to realise the design. To optimize the user experience, we have implemented the computation engine with careful treatment of the numerical computational errors, new discrete optimization strategies, and optimized compiled code.

As the result of these optimizations, the optimal design calculation can be done in around ten seconds for models that do not have an ODE and around one minute for models with stiff ODEs (EDs optimal criteria used and time measured on MacBook Pro Mid 2014 with 2.5GHz Intel Core i7 processr with 16GB of memory). Future work may include improvement of the computational speed by parallelisation, further optimisation of the ODE solvers, and refinement of the discrete optimisation algorithm.

Deployment of research results through software is not uncommon in the biomedical field and those software have the potential to bridge the gap between theoretically oriented researchers in academia and more practically oriented researchers in the industry. The development of PopED lite shows that a cross-disciplinary effort to design a purpose-specific "app" can be a powerful way of introducing a new research idea to industry.

6 Acknowledgements

Authors would like to thank Dr. Sebastian Ueckert and Dr. Joakim Nyberg for valuable and stimulating discussions, Dr. Ron Keizer for sharing his experiences in software development in pharmacometrics and suggesting the Qt framework, and the Uppsala Pharmacometrics Research Group and the AstraZeneca CVMD DMPK modeling and simulation section for useful input and software testing.

A Fisher Information Matrix

Let $f(\theta; \xi)$ be a vector function that maps from the model parameters θ to the PK and PD prediction of a given experimental design ξ . For example, for a standard one-compartment PK model with an E_{max} PD model and an intra-venous bolus administration, the parameter vector is defined as

$$\theta = [V, CL, E_{max}, ED_{50}]^T. \quad (3)$$

The design vector ξ for a design with a single dose and N_{obs} measurement points at times $t_1, t_2, \dots, t_{N_{obs}}$ is defined as

$$\xi = [t_1, t_2, \dots, t_{N_{obs}}, \text{dose amount}]^T. \quad (4)$$

Then, we let the i th element of the vector $f(\theta; \xi)$ be the PK model prediction at time t_i and $(i + N_{obs})$ th element represent the PD model prediction at time t_i (we assume both PK and PD measurements are made at the same time points).

We define the FIM as

$$FIM(\theta; \xi) := J^T(\Sigma^{-1})J \quad (5)$$

where J is a Jacobian matrix of f with respect to the model parameters θ , i.e.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \dots & \frac{\partial f_1}{\partial \theta_{N_{para}}} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \dots & \frac{\partial f_2}{\partial \theta_{N_{para}}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{2N_{obs}}}{\partial \theta_1} & \frac{\partial f_{2N_{obs}}}{\partial \theta_2} & \dots & \frac{\partial f_{2N_{obs}}}{\partial \theta_{N_{para}}} \end{bmatrix}, \quad (6)$$

and Σ is a diagonal matrix representing the prior knowledge of the measurement error and the limit of quantification. We interpret the limit of quantification as the minimum value of the residual error, for example, assuming 10 % measurement error on the PK observations, $E_{max}/10$ measurement error on the PD observations and 0.01 and 1 to be the limit of quantification of PK and PD measurements (these quantities are specified by the user in

Step 2 in the GUI), Σ becomes

$$\Sigma = \text{diag} \left(\left[\max \left(\left| \frac{f_1}{10} \right|, 0.01 \right)^2, \max \left(\left| \frac{f_2}{10} \right|, 0.01 \right)^2, \dots, \max \left(\left| \frac{f_{N_{obs}}}{10} \right|, 0.01 \right)^2, \right. \right. \\ \left. \left. \max \left(\left| \frac{E_{max}}{10} \right|, 1 \right)^2, \max \left(\left| \frac{E_{max}}{10} \right|, 1 \right)^2, \dots, \max \left(\left| \frac{E_{max}}{10} \right|, 1 \right)^2 \right] \right). \quad (7)$$

Other advanced technique to handle the limit of quantification can be found in [21] and will be implemented in a future version of the software.

If J is a full-rank matrix and all the diagonal elements of Σ are nonzero, the FIM is a non-singular matrix. In such cases, we define the predicted standard deviation of the parameter estimation uncertainty of the i th parameter (PSD_i) to be the square root of the i th diagonal element of the inverse of the FIM, i.e.

$$PSD_i(\theta; \xi) := \sqrt{((FIM(\theta; \xi))^{-1})_{i,i}}. \quad (8)$$

Finally, the predicted CV of the i th parameter (PCV_i) is defined as

$$PCV_i(\theta; \xi) := \frac{PSD_i(\theta; \xi)}{\theta_i}. \quad (9)$$

The predicted CV can be thought of as the predicted accuracy of the estimated parameter from the experiment conducted with the design ξ assuming θ is the true parameter.

Note that if the Jacobian is a full-rank matrix and the limit of quantification is set to be a positive number, then FIM is a non-singular matrix. In order to reduce the chance of FIM being numerically singular, we set the lower bound of the limit of quantification to be the square root of the machine accuracy.

Once the FIM is created, we compute the Cholesky decomposition of the FIM. If the decomposition algorithm breaks down, PopED lite annotates the FIM as numerically singular and outputs a warning that the parameter is not identifiable. If the decomposition can be computed, PopED lite rebuilds the FIM from the decomposition and compares each element to the corresponding element of the original FIM. If the maximum absolute value of the difference in an element is more than the square root of the machine accuracy, there is a significant influence of the rounding error in the computation and PopED lite outputs a warning. The decomposition is stored for future usage.

B Optimization criterion

There are several ways to quantify the goodness of the design using the FIM, see e.g. [3, 7]. PopED lite implements ED and EDs optimal designs to be the default.

B.1 Design optimization for all parameters (ED optimal design)

Consider the case when the goal of the experiment is to accurately estimate all model parameters. One alternative is to optimize for the sum (or squares of the sum) of the predicted CVs. However, for this approach the optimality of the design can be influenced by the linear reparameterization of the model (see [3] for a detailed discussion). Therefore, we choose instead to quantify the goodness of the experimental design by the determinant of the FIM, which is independent of the linear reparameterization of the model. Taking uncertainty in the prior parameter information into account, the robust optimal design is calculated as

$$\xi^* = \operatorname{argmin}_{\xi} \left(\sum_{j=1}^N (\log(\det(\operatorname{FIM}(\theta_j; \xi))) / N) \right), \quad (10)$$

where θ_j are the parameter vectors created uniformly randomly around the "Guesstimated Parameter Value" with the bound specified as "Confidence of Guesstimate" (specified in Step 2 of the GUI). Furthermore, N is the number of randomly generated parameter vectors θ_j (default is set to 25 for practical reasons; however, this setting can be changed in Step 4 of the GUI).

Note that a naive computation of the determinant of the FIM would often result in accumulation of rounding errors. In PopED lite the log of the determinant of the FIM is calculated by summing the log of the diagonal elements of its Cholesky decomposition and multiplying the sum by two. If Cholesky decomposition cannot be computed due to a singularity of the FIM, the log of the determinant of such matrices will be set to -10^{10} and not to negative infinity. In this way, PopED lite selects the experimental design with the least possibility of an unidentifiable system.

As alternatives to (10), the following robust criteria are implemented in PopED lite (can be set in Step 4 in the GUI):

$$\xi^* = \operatorname{argmin}_{\xi} \left(\min_j (\log(\det(\operatorname{FIM}(\theta_j; \xi)))) \right), \quad (11)$$

$$\xi^* = \operatorname{argmin}_{\xi} (\operatorname{median}_j (\log(\det(\operatorname{FIM}(\theta_j; \xi))))). \quad (12)$$

B.2 Design optimization for a subset of parameters (EDs optimal design)

As a second option, consider the case when a given subset of the parameters is of particular interest. The robust optimal design is then calculated as

$$\xi^* = \operatorname{argmin}_{\xi} \left(\sum_{j=1}^N \log \left(\frac{\det(\operatorname{FIM}(\theta_j; \xi))}{\det(\operatorname{FIM}_{\setminus \text{subset}}(\theta_j; \xi))} \right) / N \right), \quad (13)$$

where $\setminus \text{subset}$ denotes the complement of the subset of the parameters of particular interest (i.e., subset of parameters of no-interest). If only one parameter is chosen to be of interest, the above optimization criterion reduces to optimizing for the CV for that particular parameter. EDs optimal design is used when the options "Optimize the Design for the parameter estimation accuracy of" is set to be either "PK parameters only" or "PD parameters only" in Step 2 in the GUI. As an alternative to Eq. (13), the following criterion can be optionally used in PopED lite (Step 4 in the GUI):

$$\xi^* = \operatorname{argmin}_{\xi} \left(\sum_{j=1}^N \log (\det(\operatorname{FIM}_{\text{subset}}(\theta_j; \xi))) / N \right). \quad (14)$$

C Weighted Random Search Algorithms

In this appendix, we describe the discrete optimization algorithms we have implemented in PopED lite. To avoid local convergence and to take advantage of the discrete nature of the optimization problem, we have constructed stochastic algorithms to find an approximately optimal design in a reasonable computational time. These algorithms are made specifically to address the discrete constrained optimisation problem that appeared as a result of the design of PopED lite.

C.1 Sampling time optimization

Consider the problem of choosing N_{obs} observation points from N_{disc} discrete possible observation points. For example, if $N_{\text{obs}} = 10$ and $N_{\text{disc}} = 120$ there are over 10^{14} possible combinations and an exhaustive search is not computationally feasible. Roughly speaking, our algorithm aims to randomly search for the best design; however, for each iteration the probability of the randomly chosen observation points are adjusted based on the previous evaluation of the experimental designs. The key feature of this weighted random search algorithm is that each individual weight is defined by the combination of a pair of possible observation points, and that all weights then

can be stored in a matrix (i.e., the i th row j th column of the weight matrix represents the probability of choosing i th possible observation point given j th possible observation point is already chosen to be an observation point).

Let $\phi(D)$ be a scalar function that takes a subset of natural numbers and returns a scalar value that we aim to maximise (e.g., the subset of the natural numbers corresponds to the discrete time points in the possible observation time window and the scalar value corresponds to the determinant of the Fisher Information Matrix), i.e.

$$\phi : \mathbb{D} \rightarrow \mathbb{R}, \quad (15)$$

where \mathbb{D} is the space of all the subsets of the natural numbers between 1 and N_{disc} whose size is N_{obs} . Note that size of \mathbb{D} is $N_{\text{disc}}! / ((N_{\text{disc}} - N_{\text{obs}})! N_{\text{obs}}!)$ which grows rapidly as N_{disc} increases when N_{obs} is fixed.

All the vectors that can potentially appear as a row of the Jacobian (6), for any design, that is used to construct the FIM can be precomputed before the start of the sampling time optimization. Hence the evaluation of each experimental design does not require any PK/PD model simulation during the optimization and is hence not very computationally intensive. As a result, the computational cost of storing and checking already evaluated designs exceeds the computational cost of the redundant evaluation of the same design. Thus the weighted random search algorithm for the sampling time optimization allows the redundant evaluation of the same design.

The algorithm is defined as follows.

Pseudocode

Construct an $N_{\text{disc}} \times N_{\text{disc}}$ matrix A with all elements assigned to 1. (This matrix A stores the pairwise weights for the random search and updated after each iteration.)

For $m = 1, 2, \dots, N_{\text{iter}}$

For $l = 1, 2, \dots, N_{\text{test}}$

Reset the temporary weight matrix B to be the weight matrix A and empty the set D_l , i.e.,

$$B = A \quad (16)$$

$$D_l = \emptyset. \quad (17)$$

Uniformly randomly choose a natural number between 1 and N_{disc} and let it be i and add it to the set D_l .

Set the i th row of the temporary weight matrix B to be 0, i.e.,

$$b_{ij} = 0 \quad \text{for all } j = 1, 2, \dots, N_{\text{disc}}. \quad (18)$$

For $n = 2, 3, \dots, N_{\text{obs}}$

Randomly choose a natural number between 1 and N_{disc} with the probability

$$\frac{b_{ki}}{\sum_{k=1}^{N_{\text{disc}}} b_{ki}} \quad \text{for } k = 1, 2, \dots, N_{\text{disc}}, \quad (19)$$

and let it be i and add it to the set D_l .

Set the i th row of the temporary weight matrix B to be 0 so that the i th sampling time will not be chosen again, i.e.,

$$b_{ij} = 0 \quad \text{for all } j = 1, 2, \dots, N_{\text{disc}}. \quad (20)$$

End for

End for

Find D_m^* such that

$$\phi(D_m^*) = \max_{l=1,2,\dots,N_{\text{test}}} \phi(D_l). \quad (21)$$

Update the weight matrix A as follows

$$a_{ij} = a_{ij} + 1 \quad \text{for all } (i, j) \in D_m^* \times D_m^*. \quad (22)$$

End for

Return the best subset D^* such that

$$\phi(D^*) = \max_{m=1,2,\dots,N_{\text{iter}}} \phi(D_m^*). \quad (23)$$

The search algorithm assumes that multiple observations at the same point are not allowed. Consequently, the probability of selecting an already chosen point is assigned zero at (18) and (20). If multiple sampling for the same time point is allowed then simply omit (18) and (20) from the algorithm. If sampling from two consecutive discrete possible observation points is not allowed (e.g. since the animal needs to rest between sampling points) the rows adjacent to the i th row can be set to zero as well.

Note that the algorithm reduces to a pure random search if the weight matrix is not updated at (22). As a default, PopED lite uses the weighted random search algorithm for sampling time optimization without possibility

of multiple sampling from the same time point.

Further extension of the algorithm such as defining the weights as a tensor so that the weight given the combination of more than two observation points can be stored is possible.

C.2 Dosage optimization

Consider the problem where N_{doses} doses are given and each dose needs to be chosen between d_{\min} and d_{\max} . We first discretize the possible dose range into N_{pd} possible doses $\{d_1, d_2, d_3, \dots, d_{N_{\text{pd}}}\}$, where $d_i \neq d_j$ if $i \neq j$ and $d_{\min} \leq d_i \leq d_{\max}$. We now wish to obtain a vector of N_{doses} indices $[i, j, k, l, \dots]$ to maximise the optimization criterion when dose amount d_i is given at the first dose event, d_j is given at the second dose event, etc. Note that there are $(N_{\text{pd}})^{N_{\text{doses}}}$ possible dose combinations and that the number grows rapidly as N_{doses} increases. It is therefore often not feasible to conduct an exhaustive search for a multiple dose experiment. Instead, we randomly search through the possible dose combinations and select the best dose combination. To save computation time, this weighted random search algorithm is designed to avoid evaluation of previously evaluated designs.

Note that this optimization problem is fundamentally different from the sampling-time optimization problem, as for the sampling time optimization, we aim to find the optimal subset of indices (i.e., the order of the indices does not matter) while for the dosage optimization we aim to find a vector of indices (i.e., the order of the indices matters).

For this dosage optimization algorithm, we construct an ordered tree to store the weights. This tree is constructed by nodes storing the weights w and edges storing weight biases c . Each path of the tree represents the design of an experiment, for example the path reaches from the root to a leaf following i, j, k th edges represents the experimental design vector $[i, j, k]$ (i.e., d_i as the first dose, d_j as the second dose and d_k as the last dose). An example of the tree is depicted in Figure 6.

Weight bias c define the fundamental importance of the possible dosage that is represented as edges in the tree. For example, if there are 10 possible dose levels for each dosing time, it is wise to search for a reasonably good combination of high, medium, or low doses in a first step, and then fine tune the dose levels in a second step. Hence, in this case we put more weight bias to these three doses, and lower weight bias to the rest. By choosing the weight bias c this way, the algorithm will most probably first search through the optimal combination between high, medium and low doses and then conduct the fine tuning during the remaining search. Weight bias c remain the same throughout the search.

The variable w is a weight of the probability of choosing the dosage that is represented by the edge connected to the parent node. For example, w_{ij} (the weight stored in the node that can be reached by path i, j from the root

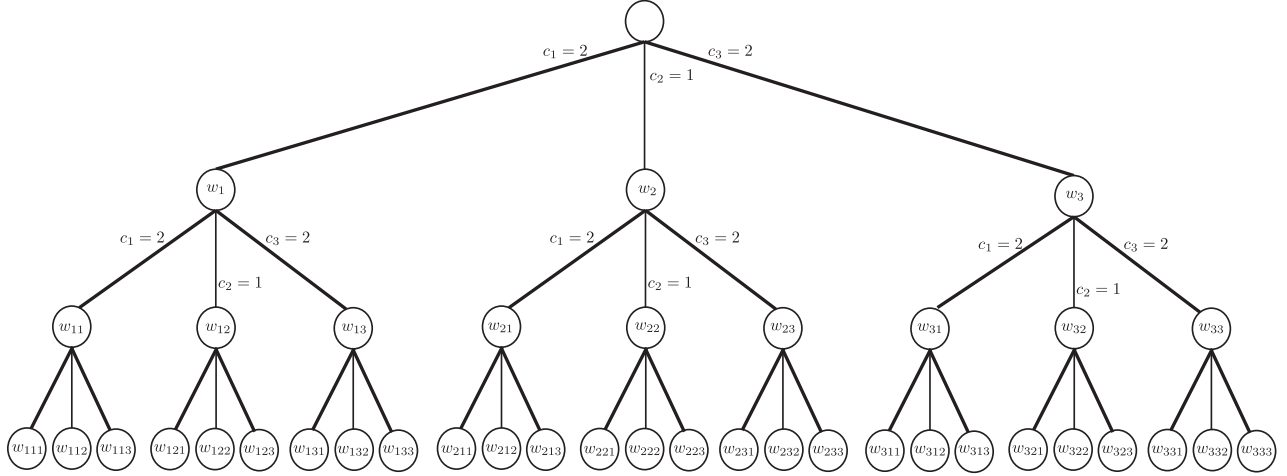


Figure 6: An example of a tree for storing the weight for the weighted random search for dosage optimisation ($N_{pd} = 2, N_{doses} = 3$). In this example, the highest and lowest possible dosages have priorities on the search and the design vectors $[111], [113], [131], [133], [311], [313], [331], [333]$ are more likely to be evaluated before other design vectors.

node) is the weight for the random choice where j th possible dosage should be chosen as a second dosage given i th possible dosage is chosen for the first dosage. The weight w depends on the weights of all the descendent nodes. For the nodes that are leaves the weights w are initially defined as weight bias c of the connecting edge and after each evaluation of the design, the weight of the corresponding leaf is reduced to zero to avoid the redundant evaluation of the same design. Hence the weights w are recalculated after each design evaluation.

We describe the algorithm more in detail using an example with $N_{doses} = 3$. The extension to other cases is trivial.

The weights of the nodes are defined as follows:

$$w_i = c_i \sum_{j=1}^{N_{pd}} w_{ij} \quad (24)$$

$$w_{ij} = c_j \sum_{k=1}^{N_{pd}} w_{ijk} \quad (25)$$

$$w_{ijk} = c_k \quad \text{for all } i, j, k. \quad (26)$$

We consider the case where we randomly choose N_{search} dose combinations and pick the best dose combination among them.

Pseudocode

For $m = 1, 2, \dots, N_{\text{search}}$

- Randomly choose an integer l between 1 and N_{pd} with the probability $w_l / \sum_{i=1}^{N_{\text{pd}}} w_i$, and let $i = l$.
- Randomly choose an integer l between 1 and N_{pd} with the probability $w_{il} / \sum_{j=1}^{N_{\text{pd}}} w_{ij}$, and let $j = l$.
- Randomly choose an integer l between 1 and N_{pd} with the probability $w_{ijl} / \sum_{k=1}^{N_{\text{pd}}} w_{ijk}$, and let $k = l$.
- Evaluate the design with dose amount d_i at the first dose time, dose amount d_j at the second dose time, and dose amount d_k at the third dose time. If this design is better than the one found previously keep this design, else discard.
- Let $w_{ijk} = 0$ so that the dose combination of ijk will not be redundantly evaluated during the following search.
- Recalculate the weights following (24)-(25).

End for

Return the best design.

Note that, the weight calculation needs to be implemented recursively so that the weight can be calculated for any N_{doses} . In addition, to reduce the computational cost of constructing the tree, we construct only the necessary part of the tree to calculate the new weights after each search iteration. Note that this algorithm avoids the redundant evaluation of the design; hence if N_{search} is sufficiently large for given N_{doses} and N_{pd} , the search is exhaustive. In addition, avoiding redundant evaluation of the same design is crucial for the dosage optimisation as each evaluation of the design requires PK/PD model evaluation and can be computationally intensive.

The weight bias c of this algorithm is set to one in the current version of PopED lite. Future work may include changing c dynamically throughout the search.

C.3 Simultaneous optimization of dose and sampling time

To optimize both dose and sampling time, conduct sampling time optimization at each search iteration in the dose optimization.

D Matrix/Vector Notations

In this paper we denote a scalar quantity with a lower case letter e.g., a or c , a matrix with a capital letter, for example A , or M , and a vector quantity by a bold symbol of a lower case letter, e.g., \mathbf{v} , \mathbf{a} , unless otherwise specifically stated. As an exception to this matrix notation, Fisher Information Matrix is denoted as FIM to follow the convention of the notation in the field. In addition the i th row of the matrix M can be denoted by a vector $\mathbf{m}_{i\cdot}$, the j th row of the matrix M can be denoted by a vector $\mathbf{m}_{\cdot j}$ and the i th row j th column of the matrix M can be written by a scalar m_{ij} .

References

- [1] K. Ahnert and M. Mulansky. Odeint - Solving ordinary differential equations in C++. In *AIP Conference Proceedings*, volume 1389, pages 1586–1589, 2011.
- [2] Apple. Instruments, 2014.
- [3] A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs*. Oxford University Press, Oxford, 1992.
- [4] C. Bazzoli, S. Retout, and F. Mentré. Design evaluation and optimisation in multiple response nonlinear mixed effect models: PFIM 3.0. *Computer Methods and Programs in Biomedicine*, 98:55–65, 2010.
- [5] Digia. Qt Project, 2014.
- [6] E.W. Dijkstra. Algol 60 translation : An algol 60 translator for the x1 and making a translator for algol 60. Technical Report MR 34/61, Stichting Mathematisch Centrum, 1961.
- [7] V. V. R. Fedorov. *Theory of optimal experiments*. Academic Press, New York, 1972.
- [8] V. V. R. Fedorov, Gagnon, S. Leonov, Y. Wu, A. Dmitrienko, C. Chuang-Stein, and R. D’Agostino. Optimal design of experiments in pharmaceutical applications. In *Pharmaceutical Statistics Using SAS. A Practical Guide*, pages 151–195. 2007.
- [9] J. Gabrielsson and D. Weiner. *Pharmacokinetic and Pharmacodynamic, Data analysis, Concepts and Applications*. Swedish Pharmaceutical Press, Stockholm, 2010.
- [10] R. L. Lalonde, K. G. Kowalski, M. M. Hutmacher, W. Ewy, D. J. Nichols, P. A. Milligan, B. W. Corrigan, P. A. Lockwood, S. A. Marshall, L. J. Benincosa, T. G. Tensfeldt, K. Parivar, M. Amantea, P. Glue, H. Koide, and R. Miller. Model-based drug development. *Clinical pharmacology and therapeutics*, 82(1):21–32, 2007.
- [11] R. Lledo-Garcia, S. Hennig, J. Nyberg, A.C. Hooker, and M.O. Karlsson. Ethically Attractive Dose-Finding Designs for Drugs With a Narrow Therapeutic Index. *The Journal of Clinical Pharmacology*, 52(1):29–38, 2012.
- [12] F. Mentré. Optimal design in random-effects regression models. *Biometrika*, 84:429–442, 1997.
- [13] J. Nyberg, M. O. Karlsson, and A.C. Hooker. Simultaneous optimal experimental design on dose and sample times. *Journal of Pharmacokinetics and Pharmacodynamics*, 36:125–145, 2009.

- [14] J. Nyberg, S. Ueckert, E. A. Strömberg, S. Hennig, M. O. Karlsson, and A. C. Hooker. PopED: An extended, parallelized, nonlinear mixed effects models optimal design tool. *Computer Methods and Programs in Biomedicine*, 108:789–805, 2012.
- [15] Z. Qi, I. Whitt, A. Mehta, J. Jin, M. Zhao, R. C. Harris, A. B. Fogo, and M. D. Breyer. Serial determination of glomerular filtration rate in conscious mice using FITC-inulin clearance. *American journal of physiology. Renal physiology*, 286:F590–F596, 2004.
- [16] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, 5:329–330, 1963.
- [17] E. Sjögren, J. Nyberg, M. O. Magnusson, H. Lennernas, A. C. Hooker, and U. Bredberg. Optimal experimental design for assessment of enzyme kinetics in a drug discovery screening environment. *Drug metabolism and disposition the biological fate of chemicals*, pages 195–210, 2011.
- [18] E. Sjögren, P. Svanberg, and K. P. Kanebratt. Optimized experimental design for the estimation of enzyme kinetic parameters: An experimental evaluation. *Drug Metabolism and Disposition*, 40:2273–2279, 2012.
- [19] C. Sturgeon, A. D. Sam, and W. R. Law. Rapid determination of glomerular filtration rate by single-bolus inulin: a comparison of estimation analyses. *Journal of applied physiology (Bethesda, Md. : 1985)*, 84:2154–2162, 1998.
- [20] S. A. G. Visser, M. Aurell, R. D. O. Jones, V. J. A. Schuck, A. C. Egnell, A. Sheila Peters, L. Brynne, J. W. T. Yates, R. Jansson-Löfmark, B. Tan, M. Cooke, S. T. Barry, A. Hughes, and U. Bredberg. Model-based drug discovery: Implementation and impact. *Drug Discovery Today*, 18(15-16):764–775, 2013.
- [21] C. Vong, S. Ueckert, J. Nyberg, and A. C. Hooker. Handling Below Limit of Quantification Data in Optimal Trial Design. *Journal of Pharmacokinetics and Pharmacodynamics*, submitted, 2015.